**АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ (ФИЛИАЛ) СПбГУТ**

**(АКТ (ф) СПбГУТ)**

**Отчеты по практическим работам**

**по ОПБД**

Студент: Д.А. Капитанская

Группа: ИСПП-21

Преподаватель: Ю.С. Маломан

Архангельск 2024

# Практическая работа №1

**Изучение правил преобразования ER-модели в реляционную модель данных**

1. **Цель работы**
   1. Изучить правила преобразования ER-модели в реляционную модель данных с учетом вида связи;
   2. Изучить процесс создания концептуальной модели данных и ее преобразования в логическую модель данных;
   3. Научиться применять графические редакторы в процессе проектирования БД.
2. **Контрольные вопросы**
   1. Сущность – объект, информацию о котором нужно хранить в БД.
   2. Атрибут – характеристика сущности, иногда связи, моделирующая ее свойства.
   3. Ключевое поле – это поле, которое содержит уникальные для каждой записи данные.
   4. Назначение первичных ключей – идентифицировать записи в таблице, а назначение внешних ключей – устанавливать связи между таблицами, ссылаясь на первичные ключи других таблиц.
   5. Связь – это логическая ассоциация между двумя или более сущностями, которая отображает отношения между ними.
   6. Виды связей между сущностями:
      1. один к одному (1:1),
      2. один ко многим (1:M),
      3. многие к одному (M:1),
      4. многие ко многим (M:M).
   7. В ER-диаграммы входят сущности, атрибуты, ключевые поля и связи.
3. **Вывод**
   1. В ходе проделанной лабораторной работы были изучены правила преобразования ER-модели в реляционную модель данных с учетом вида связи;
   2. В ходе проделанной лабораторной работы был изучен процесс создания концептуальной модели данных и ее преобразования в логическую модель данных;
   3. В ходе проделанной лабораторной работы было сформировано умение применять графические редакторы в процессе проектирования БД.

# Практическая работа №2

**Изучение процесса нормализации отношений**

1. **Цель работы**
   1. Изучить процесс приведения отношений от ненормализованного вида к четвертой нормальной форме;
   2. Изучить процесс декомпозиции отношений.
2. **Контрольные вопросы**
   1. Первичный ключ отношения – это атрибут или набор атрибутов, которые идентифицируют каждую сущность.
   2. Внешний ключ отношения – это атрибут или набор атрибутов, которые ссылаются на первичный ключ другой сущности, устанавливая связь между ними.
   3. Нормализация отношений – это процесс разбиения отношения на более мелкие отношения, чтобы устранить избыточность данных и обеспечить их целостность.
   4. Атрибут В функционально зависит от атрибута А, если каждому значению А соответствует в точности одно значение В.
   5. Атрибут А транзитивно зависит от атрибута B, если он зависит от другого атрибута C, который, в свою очередь, зависит от B.
   6. Отношение находится в 1НФ, если в каждой ячейке есть только одно значение.
   7. Отношение находится во 2НФ, если оно находится в 1НФ и каждый ее ключевой атрибут находится в полной функциональной зависимости.
   8. Отношение находится в 3НФ, если оно находится во 2НФ и не имеет транзитивной зависимости между ключевыми и неключевыми атрибутами.
3. **Вывод**
   1. В ходе проделанной практической работы был изучен процесс приведения отношений от ненормализованного вида к четвертой нормальной форме;
   2. В ходе проделанной практической работы был изучен процесс декомпозиции отношений.

# Практическая работа №3

**Разработка физической модели данных**

1. **Цель работы**
   1. Изучить процесс создания физической модели данных.
2. **Контрольные вопросы**
   1. Физическая модель данных — это реализация логической модели данных, создаваемая администраторами и разработчиками баз данных.
   2. Для хранения строковых значений используются следующие типы данных: varchar(длина), nvarchar(длина), char(длина), nchar(длина).
   3. Для хранения целочисленных значений используются следующие типы данных: int, tinyint, bigint.
   4. Для хранения вещественных значений используется тип данных decimal.
   5. Для хранения логических значений используется тип данных bit.
   6. Для хранения даты и/или времени используются следующие типы данных: date, datetime.
   7. Ограничение целостности – это свойство, которое для данного множества или отношения либо истинно, либо ложно.
   8. Виды ограничений:
      1. ограничение первичного ключа,
      2. ограничение внешнего ключа,
      3. уникальность одного или набора столбцов,
      4. проверочное ограничение,
      5. значение по умолчанию.
3. **Вывод**
   1. В ходе проделанной практической работы был изучен процесс создания физической модели данных.

# Практическая работа №4

**Разработка ER-модели БД в Microsoft Visio**

1. **Цель работы**
   1. Изучить процесс создания концептуальной, логической и физической модели данных в MS Visio.
2. **Контрольные вопросы**
   1. MS Visio – это программное обеспечение для создания диаграмм и визуализации, которое можно использовать для проектирования схем баз данных.
   2. В MS Visio схемы баз данных можно проектировать в следующих нотациях:
      1. нотация «воронья лапка»,
      2. нотация Айзенбаха,
      3. нотация Баркера,
      4. IDEF1X,
      5. UML-диаграммы классов.
   3. Для настройки схемы БД в MS Visio доступны следующие элементы:
      1. таблицы,
      2. столбцы,
      3. связи,
      4. индексы,
      5. ограничения,
      6. триггеры,
      7. хранимые процедуры.
   4. Чтобы задать первичный ключ в MS Visio, нужно выбрать столбец таблицы и нажать кнопку “Ключ” на панели инструментов “Инструменты базы данных”. Затем выбрать “Первичный ключ” в раскрывающемся меню.
   5. Чтобы задать внешний ключ в MS Visio, нужно выбрать столбец таблицы и нажать кнопку “Связь” на панели инструментов “Инструменты базы данных”. Затем выбрать “Внешний ключ” в раскрывающемся меню и указать таблицу и столбец, на которые ссылается внешний ключ.
   6. Чтобы изменить тип связи в MS Visio, нужно выбрать связь и нажать кнопку “Тип связи” на панели инструментов “Инструменты базы данных”. Затем выбрать тип связи (1:1, М:М или 1:М) в раскрывающемся меню.
   7. Чтобы изменить связь на идентифицирующую или не идентифицирующую в MS Visio, нужно выбрать связь и нажать кнопку “Идентифицирующая связь” на панели инструментов “Инструменты базы данных”. Затем выбрать “Идентифицирующая связь” или “Не идентифицирующая связь” в раскрывающемся меню.
3. **Вывод**
   1. В ходе проделанной практической работы был изучен процесс создания концептуальной, логической и физической модели данных в MS Visio.

# Практическая работа №5

**Разработка ER-модели БД в MySQL Workbench**

1. **Цель работы**
   1. Научиться применять MySQL Workbench в процессе создания схем моделей БД;
   2. Научиться представлять логическую модель данных согласно нотациям ERD и IDEF1X.
2. **Контрольные вопросы**
   1. Сущность – объект, информацию о котором нужно хранить в БД.
   2. Атрибут – характеристика сущности, иногда связи, моделирующая ее свойства.
   3. Ключевое поле – это поле, которое содержит уникальные для каждой записи данные.
   4. Назначение первичных ключей –- идентифицировать записи в таблице, а назначение внешних ключей – устанавливать связи между таблицами, ссылаясь на первичные ключи других таблиц.
   5. Связь – это логическая ассоциация между двумя или более сущностями, которая отображает отношения между ними.
   6. Виды связей между сущностями:
      1. один к одному (1:1),
      2. один ко многим (1:M),
      3. многие к одному (M:1),
      4. многие ко многим (M:M).
   7. В ER-диаграммы входят сущности, атрибуты, ключевые поля и связи.
   8. ER-диаграммы применяются для анализа уже имеющихся баз данных с целью выявить и устранить ошибки в логике или развертывании.
3. **Вывод**
   1. В ходе проделанной практической работы было сформировано умение применять MySQL Workbench в процессе создания схем моделей БД;
   2. В ходе проделанной практической работы было сформировано умение представлять логическую модель данных согласно нотациям ERD и IDEF1X.

# Практическая работа №6

**Создание БД MySQL**

1. **Цель работы**
   1. Научиться создавать физическую модель данных БД MySQL, используя CASE-средство MySQL Workbench.
   2. Научиться создавать БД MySQL, используя CASE-средство MySQL.
2. **Контрольные вопросы**
   1. MySQL Workbench – это интегрированная среда разработки для управления базами данных MySQL. Она используется для разработки, моделирования, управления и администрирования баз данных MySQL.
   2. Чтобы создать EER-модель в MySQL Workbench, нужно выполнить следующие действия:
      1. Открыть MySQL Workbench;
      2. Нажать кнопку “Создать модель” на панели инструментов;
      3. Выбрать тип модели “EER-модель”;
      4. Ввести имя модели;
      5. Добавить сущности, атрибуты и связи на диаграмму;
      6. Сохранить модель.
   3. EER-модель состоит из сущностей, атрибутов и связей.
   4. Прямой инжиниринг – это процесс создания модели базы данных на основе существующей базы данных.
   5. Чтобы провести прямой инжиниринг, используя MySQL Workbench, нужно выполнить следующие действия:
      1. Открыть MySQL Workbench;
      2. Подключиться к существующей базе данных MySQL;
      3. Щелкнуть правой кнопкой мыши на базе данных и выберите “Reverse Engineer”;
      4. Выбрать тип модели “EER-модель”;
      5. Ввести имя модели;
      6. Выбрать опции для прямого инжиниринга;
      7. Нажать “OK”.
3. **Вывод**
   1. В ходе проделанной практической работы было сформировано умение создавать физическую модель данных БД MySQL, используя CASE-средство MySQL Workbench;
   2. В ходе проделанной практической работы было сформировано умение создавать БД MySQL, используя CASE-средство MySQL.

# Практическая работа №7

**Изучение способов обеспечения целостности данных в СУБД MySQL**

1. **Цель работы**
   1. Изучить способы обеспечения целостности данных в БД MySQL Server, используя CASE-средство MySQL Workbench.
2. **Контрольные вопросы**
   1. Чтобы указать автоинкрементные значения в MySQL Workbench, нужно выполнить следующие действия:
      1. Щелкнуть правой кнопкой мыши на таблице и выбрать “Изменить таблицу”;
      2. В разделе “Столбцы” найти столбец, для которого нужно указать автоинкремент;
      3. Установить флажок “Автоинкремент”.
   2. Чтобы указать значения по умолчанию в MySQL Workbench, нужно выполнить следующие действия:
      1. Щелкнуть правой кнопкой мыши на таблице и выбрать “Изменить таблицу”;
      2. В разделе “Столбцы” найтм столбец, для которого нужно указать значение по умолчанию;
      3. В поле “Значение по умолчанию” введите значение по умолчанию.
   3. Чтобы указать обязательные для заполнения столбцы в MySQL Workbench нужно выполнить следующие действия:
      1. Щелкнуть правой кнопкой мыши на таблице и выбрать “Изменить таблицу”;
      2. В разделе “Столбцы” найти столбец, который нужно сделать обязательным для заполнения;
      3. Установить флажок “Не допускать значения NULL”.
   4. Чтобы создать индексы в MySQL Workbench, нужно выполнить следующие действия:
      1. Щелкнуть правой кнопкой мыши на таблице и выбрать “Создать индекс”;
      2. В диалоговом окне “Создание индекса” ввести имя индекса;
      3. Выбрать тип индекса;
      4. Выбрать столбцы, которые будут включены в индекс.
   5. Виды индексов, которые можно создать в таблицах MySQL:
      1. уникальный индекс,
      2. неуникальный индекс,
      3. индекс полнотекстового поиска,
      4. пространственный индекс.
   6. Чтобы указать настройки внешних ключей в MySQL Workbench, нужно выолнить следующие действия:
      1. Щелкнуть правой кнопкой мыши на таблице и выберите “Изменить таблицу”;
      2. В разделе “Связи” найти столбец внешнего ключа;
      3. Нажать кнопку “Изменить” и выбрать таблицу и столбец первичного ключа, на который ссылается внешний ключ.
   7. Чтобы указать уникальность значений в одном столбце в MySQL Workbench, нужно выполнить следующие действия:
      1. Щелкнуть правой кнопкой мыши на таблице и выбрать “Изменить таблицу”;
      2. В разделе “Столбцы” найти столбец, для которого нужно указать уникальность;
      3. Установить флажок “Уникальный”.
   8. Чтобы указать уникальность значений в наборе столбцов в MySQL Workbench, нужно выполнить следующие действия:
      1. Щелкнуть правой кнопкой мыши на таблице и выбрать “Изменить таблицу”;
      2. В разделе “Ключи” нажать кнопку “Создать”;
      3. Выбрать тип ключа “Уникальный”;
      4. Выбрать столбцы, которые будут включены в ключ.
3. **Вывод**
   1. В ходе проделанной практической работы были изучены способы обеспечения целостности данных в БД MySQL Server, используя CASE-средство MySQL Workbench.

# Практическая работа №8

# Создание БД Microsoft SQL Server

1. **Цель работы**
   1. Изучить процесс создания таблиц и связей между ними в СУБД Microsoft SQL Server (MSSQL);
   2. Научиться работать в среде SQL Server Management Studio (SSMS).
2. **Контрольные вопросы**
   1. SQL Server Management Studio – это интегрированная среда разработки для управления базами данных Microsoft SQL Server. Она позволяет пользователям создавать, редактировать, отлаживать и администрировать базы данных SQL Server.
   2. Виды авторизации, поддерживаемые в MS SQL Server:
      1. авторизация Windows,
      2. SQL Server-авторизация.
   3. Чтобы создать диаграмму в SQL Server Management Studio, нужно выполнить следующие действия:
      1. Открыть SQL Server Management Studio;
      2. Подключиться к базе данных SQL Server;
      3. Щелкнуть правой кнопкой мыши на базе данных и выбрать “Новая диаграмма”;
      4. В диалоговом окне “Новая диаграмма” выбрать тип диаграммы;
      5. Ввести имя диаграммы;
      6. Нажать кнопку “OK”.
   4. Чтобы добавить таблицы в диаграмму в SQL Server Management Studio, нужно выполнить следующие действия:
      1. Открыть диаграмму;
      2. Перетащить таблицы из обозревателя объектов на диаграмму;
      3. Отпустить кнопку мыши, чтобы добавить таблицы в диаграмму.
   5. Чтобы создать связи между таблицами в SQL Server Management Studio, нужно выполнить следующие действия:
      1. Открыть диаграмму;
      2. Щелкнуть правой кнопкой мыши на таблице и выбрать “Связи”;
      3. В диалоговом окне “Свойства связи” указать связь между таблицами;
      4. Нажать кнопку “OK”.
3. **Вывод**
   1. В ходе проделанной практической работы был изучен процесс создания таблиц и связей между ними в СУБД Microsoft SQL Server (MSSQL);
   2. В ходе проделанной практической работы было сформировано умение работать в среде SQL Server Management Studio (SSMS).

# Практическая работа №9

**Изучение способов обеспечения целостности данных** **в СУБД Microsoft SQL Server**

1. **Цель работы**
   1. Изучить способы обеспечения целостности данных в БД Microsoft SQL Server, используя SQL Server Management Studio (SSMS).
2. **Контрольные вопросы**
   1. Чтобы указать автоинкрементные значения в SSMS, нужно выполнить следующие действия:
      1. Щелкнуть правой кнопкой мыши на таблице и выбрать “Изменить столбец”;
      2. В столбце “Значение по умолчанию” ввести “IDENTITY(1, 1)”.
   2. Чтобы указать значения по умолчанию в SSMS, нужно выполнить следующие действия:
      1. Щелкнуть правой кнопкой мыши на таблице и выбрать “Изменить столбец”;
      2. В столбце “Значение по умолчанию” ввести значение, которое должно быть по умолчанию для этого столбца.
   3. Чтобы указать необязательные для заполнения столбцы в SSMS, нужно выполнить следующие действия:
      1. Щелкнуть правой кнопкой мыши на таблице и выбрать “Изменить столбец”;
      2. В столбце “Разрешать значения NULL” установить флажок “Да”.
   4. Чтобы указать настройки внешних ключей в SSMS, нужно выполнить следующие действия:
      1. Щелкнуть правой кнопкой мыши на таблице и выбрать “Изменить связи”;
      2. В диалоговом окне “Изменение связи” выбрать столбец, который будет внешним ключом;
      3. В столбце “Столбец внешнего ключа” выбрать столбец, на который ссылается внешний ключ;
      4. Установить флажок “Обеспечение целостности ссылок”.
   5. Чтобы указать уникальность значений в SSMS, нужно выполнить следующие действия:
      1. Щелкнуть правой кнопкой мыши на таблице и выбрать “Изменить индексы и ключи”;
      2. В диалоговом окне “Индексы и ключи” нажать кнопку “Добавить”;
      3. В столбце “Имя” ввести имя индекса;
      4. В столбце “Столбец” выбрать столбец, для которого нужно указать уникальность;
      5. Установить флажок “Уникальный”.
3. **Вывод**
   1. В ходе проделанной практической работы были изучены способы обеспечения целостности данных в БД Microsoft SQL Server, используя SQL Server Management Studio (SSMS).

# Практическая работа №10

**Создание SQL-запросов на модификацию схемы БД**

1. **Цель работы**
   1. Изучить процесс разработки DDL-команд в БД Microsoft SQL Server, используя SQL Server Management Studio (SSMS).
2. **Контрольные вопросы**
   1. Предложение PRIMARY KEY позволяет определить внешний ключ.
   2. Предложение FOREIGN KEY позволяет определить внешний ключ.
   3. Ограничения задаются в предложении CONSTRAINT.
   4. Предложение CONSTRAINT используется для создания ограничений.
   5. Столбец нельзя удалить, если он участвует в первичном или внешнем ключе. Таблицу нельзя удалить, если она ссылается на нее по внешнему ключу.
   6. Команда ALTER TABLE предназначена для изменения структуры существующей таблицы.
   7. Команда DROP TABLE предназначена для удаления таблицы и всех ее данных.
3. **Вывод**
   1. В ходе проделанной практической работы были изучены способы обеспечения целостности данных в БД Microsoft SQL Server, используя SQL Server Management Studio (SSMS).

# Практическая работа №11

**Создание SQL-запросов на модификацию данных**

1. **Цель работы**
   1. Изучить процесс создания таблиц и связей между ними в Microsoft SQL Server (MSSQL);
   2. Научиться работать в среде SQL Server Management Studio (SSMS).
2. **Контрольные вопросы**
   1. Причины ошибок, связанных с выполнением DML-команд на модификацию данных, могут включать:
      1. Нарушение ограничений целостности: например, попытка вставить дублирующийся ключ или значение, которое нарушает ограничение уникальности.
      2. Ошибки синтаксиса: неправильное использование ключевых слов или неверный порядок элементов в команде.
      3. Проблемы с доступом к данным: отсутствие прав на изменение данных или попытка изменить данные, которые в данный момент блокируются другой транзакцией.
      4. Несоответствие типов данных: попытка вставить данные, тип которых не соответствует типу столбца в таблице.
   2. В запросе на обновление после ключевого слова SET записывается обновляемое поле.
   3. Команда INSERT используется для добавления новых записей в таблицу базы данных.
   4. Команда UPDATE используется для изменения существующих записей в таблице базы данных.
   5. Команда DELETE используется для удаления записей из таблицы базы данных.
3. **Вывод**
   1. В ходе проделанной практической работы был изучен процесс создания таблиц и связей между ними в Microsoft SQL Server (MSSQL) и сформировано умение работать в среде SQL Server Management Studio (SSMS).

# Практическая работа №12

**Создание однотабличных SQL-запросов на выборку данных**

1. **Цель работы**
   1. Изучить процесс создания однотабличных SQL-запросов на выборку.
2. **Контрольные вопросы**
   1. Структура инструкции SELECT обычно следующая:
      1. SELECT столбцы,
      2. FROM таблица,
      3. WHERE условия,
      4. GROUP BY группировка,
      5. HAVING условия\_группировки,
      6. ORDER BY сортировка,
      7. LIMIT ограничение\_количества.
   2. После ключевого слова SELECT указываются столбцы или выражения, которые необходимо извлечь из таблицы.
   3. После ключевого слова FROM указывается имя таблицы, из которой будут извлекаться данные.
   4. После ключевого слова GROUP BY указываются столбцы, по которым будет производиться группировка результатов запроса.
   5. После ключевого слова ORDER BY указываются столбцы, по которым будет производиться сортировка результатов запроса, а также направление сортировки (ASC для возрастания или DESC для убывания).
3. **Вывод**
   1. В ходе проделанной практической работы был изучен процесс создания однотабличных SQL-запросов на выборку.

# Практическая работа №13

**Создание однотабличных SQL-запросов на выборку с фильтрацией данных**

1. **Цель работы**
   1. Изучить процесс фильтрации данных в SQL-запросах на выборку.
2. **Контрольные вопросы**
   1. После ключевого слова WHERE указываются условия, которые фильтруют результаты запроса. Эти условия могут включать сравнения, логические операторы и другие функции, ограничивающие строки, возвращаемые запросом.
   2. После ключевого слова HAVING указываются условия, которые фильтруют группы, созданные оператором GROUP BY. Эти условия применяются к агрегированным данным, таким как суммы, средние значения и т.д.
   3. Отличие между разделами WHERE и HAVING заключается в том, что WHERE фильтрует строки до группировки, а HAVING фильтрует группы после группировки. WHERE не может использоваться с агрегатными функциями, в то время как HAVING предназначен именно для этого.
   4. После ключевого слова IN указывается список значений, с которыми будет сравниваться столбец или выражение. Это позволяет фильтровать результаты запроса на основе соответствия любому из значений в списке.
   5. После ключевого слова LIKE указывается шаблон для сравнения строк. Шаблон может включать специальные символы, такие как % (заменяет любую последовательность символов) и \_ (заменяет один символ), для выполнения поиска по шаблону.
   6. После ключевого слова BETWEEN указываются два значения, которые определяют диапазон. Это условие используется для фильтрации результатов запроса, чтобы включить только те строки, где столбец или выражение находится в указанном диапазоне значений.
3. **Вывод**
   1. В ходе проделанной практической работы был изучен процесс фильтрации данных в SQL-запросах на выборку.

# Практическая работа №14

**Создание многотабличных SQL-запросов на выборку данных**

1. **Цель работы**
   1. Изучить процесс создания многотабличных SQL-запросов на выборку.
2. **Контрольные вопросы**
   1. После ключевого слова FROM указывается имя таблицы или нескольких таблиц, из которых будут извлекаться данные.
   2. Синтаксис инструкции объединения двух таблиц выглядит следующим образом:
      1. SELECT столбцы,
      2. FROM таблица1,
      3. JOIN таблица2 ON таблица1.столбец = таблица2.столбец;
   3. Существуют следующие виды объединения таблиц:
      1. INNER JOIN: возвращает строки, когда есть совпадение в обеих таблицах.
      2. LEFT JOIN (или LEFT OUTER JOIN): возвращает все строки из левой таблицы и совпадающие строки из правой таблицы.
      3. RIGHT JOIN (или RIGHT OUTER JOIN): возвращает все строки из правой таблицы и совпадающие строки из левой таблицы.
      4. FULL JOIN (или FULL OUTER JOIN): возвращает строки, когда есть совпадение хотя бы в одной из таблиц.
   4. Объединение LEFT JOIN возвращает все строки из левой (первой) таблицы, даже если нет совпадений в правой (второй) таблице. В случае RIGHT JOIN, возвращаются все строки из правой (второй) таблицы, даже если нет совпадений в левой (первой) таблице.
   5. Подзапрос — это запрос внутри другого запроса. Он может быть использован в различных частях основного запроса, таких как SELECT, FROM, WHERE и т.д., для выполнения вложенного запроса и возврата данных, которые затем используются в основном запросе.
   6. Ключевое слово UNION используется для объединения результатов двух или более запросов SELECT в один набор результатов, включая только уникальные строки. Все запросы SELECT внутри UNION должны иметь одинаковое количество столбцов в результатах с совместимыми типами данных.
3. **Вывод**
   1. В ходе проделанной практической работы был изучен процесс создания многотабличных SQL-запросов на выборку.

# Практическая работа №15

**Применение встроенных функций SQL**

1. **Цель работы**
   1. Изучить процесс создания SQL-запросов, содержащих встроенные функции в СУБД MS SQL Server.
2. **Контрольные вопросы**
   1. Преимущества применения встроенных функций включают:
      1. Эффективность: Встроенные функции оптимизированы для быстрого выполнения, что ускоряет обработку запросов.
      2. Надежность: Они тщательно протестированы и являются надежными, что снижает вероятность ошибок в коде.
      3. Удобство: Упрощают написание кода, так как не требуют создания пользовательских функций для стандартных операций
      4. Портируемость: Стандартизированы и поддерживаются многими СУБД, что облегчает перенос кода между системами.
   2. В MSSQL существуют следующие виды системных функций:
      1. Функции агрегирования: такие как SUM(), COUNT(), AVG(), предназначены для выполнения операций над набором значений и возвращения одного значения.
      2. Скалярные функции: возвращают одно значение на основе входных параметров, например, LEN(), GETDATE(), SUBSTRING().
      3. Функции ранжирования: такие как ROW\_NUMBER(), RANK(), позволяют присваивать ранги строкам в результирующем наборе.
      4. Функции оконные: работают с набором строк, определенным окном, например, LEAD(), LAG(), FIRST\_VALUE().
      5. Функции системной информации: предоставляют информацию о системе, например, @@VERSION, @@IDENTITY.
   3. Параметры в функцию указываются в скобках после имени функции, разделяя их запятыми. Например, функция SUBSTRING() принимает три параметра: SUBSTRING(строка, начало, длина). Если функция не требует параметров, используются пустые скобки: GETDATE().
3. **Вывод**
   1. В ходе проделанной практической работы был изучен процесс создания SQL-запросов, содержащих встроенные функции в СУБД MS SQL Server.